



# WavLM model ensemble for audio deepfake detection

David Combei,<sup>1</sup> Adriana Stan,<sup>1,2</sup> Dan Oneată,<sup>2</sup> Horia Cucu<sup>2</sup>

<sup>1</sup>Technical University of Cluj-Napoca, Romania

<sup>2</sup>POLITEHNICA Bucharest, Romania

combei.ga.david@student.utcluj.ro, adriana.stan@com.utcluj.ro

dan.oneata@gmail.com, horia.cucu@upb.ro

## Abstract

Audio deepfake detection has become a pivotal task over the last couple of years, as many recent speech synthesis and voice cloning systems generate highly realistic speech samples, thus enabling their use in malicious activities. In this paper we address the issue of audio deepfake detection as it was set in the ASVspoof5 challenge. First, we benchmark ten types of pretrained representations and show that the self-supervised representations stemming from the wav2vec2 and wavLM families perform best. Of the two, wavLM is better when restricting the pretraining data to LibriSpeech, as required by the challenge rules. To further improve performance, we finetune the wavLM model for the deepfake detection task. We extend the ASVspoof5 dataset with samples from other deepfake detection datasets and apply data augmentation. Our final challenge submission consists of a late fusion combination of four models and achieves an equal error rate of 6.56% and 17.08% on the two evaluation sets.

## 1. Introduction

The capacity of generative deep learning has recently achieved remarkable results, and it has become close to impossible to perceptually distinguish between real (or *bonafide*) and generated (or *fake*, *spoofed*) data across multiple domains. Audio generation is no exception. High-quality text-to-speech (TTS) and voice cloning (VC) systems have become easily and readily available for all user categories. If the use of such technology is performed on one's behalf, for example to generate audio content for a video blog or online platform, its use supports the user tremendously and eases the process of generating vast amounts of online content. However, if these systems are used to impersonate or to alter an original audio or video resource, then the forensics of deepfake data should be supported by equally able detection systems.

**Audio deepfake detection.** Modern deepfake detection increasingly relies on self-supervised representations [1–8]. Self-supervised learning (SSL) [9] is a powerful paradigm that aims to produce transferable representations. Methods such as wav2vec [10], HuBERT [11] or wavLM [12] achieve this desideratum by reconstructing masked parts of the input audio. The resulting representations can be successfully employed by multiple downstream tasks (e.g. speech recognition, keyword spotting, speaker identification) with limited data [13]. This is also the case for our task of interest, audio deepfake detection, where methods based on self-supervised representations provide a lighter [7] and more robust [8] alternative compared to the previous generation of approaches, such as ResNet on linear frequency cepstral coefficients (LFCC) [14] or RawNet [15].

Most approaches use self-supervised models as feature ex-

tractors, keeping them frozen during training [2, 4, 6–8]. Finetuning the self-supervised frontends has also been explored [1, 3, 5], although to a lesser degree; among these, Wang and Yamagishi [1] suggest that finetuning the frontend reduces the reliance of the classifier on the type of backend. The wav2vec family of models [10, 16, 17] remains the most popular option for deepfake detection [1–3, 7, 8, 18, 19]. WavLM and HuBERT representations have also been employed [1, 4, 5, 8], but their results seem to trail those of wav2vec. Wav2vec comes in multiple variants and while most of the small versions have been employed [1, 7], the larger variants trained on more data perform best [8]. Others have also combined these three types of features [4] or used representations from earlier layers [7] or pooled information from multiple layers [2, 5, 6].

Backends range from simple linear models [7, 8] to more complex pooling mechanisms [1, 2, 5, 6]. At one end of the spectrum, Pascu et al. [8] and Saha et al. [7] observe good results even for linear classifiers. At the other end of the spectrum, Wang and Yamagishi [1] suggest that more complex backends help more, with multi-fusion attention mechanism being a popular choice [5, 6].

**Our work.** In this paper we address the topic of unimodal audio deepfake detection (or spoofing) in the context of the 2024 ASVspoof5 Challenge (ASV5) [20]. The challenge was based on a very large crowdsourced dataset of spoofed audio samples generated with various TTS and VC systems. It contained two tracks: 1) deepfake detection; 2) automatic speaker verification. For both tracks, closed and open conditions were also in place. The closed condition referred to using only the released data, thus restricting the use of other spoken samples or pretrained models. In the open condition, there was a single limitation pertaining to the use of models or datasets which included samples from the LibriLight [21], Multilingual LibriSpeech [22] or MUSAN [23] datasets.

Our challenge submissions and results address the *open condition of the deepfake detection track*. As shown above, large SSL models have shown very good performance over the deepfake detection task. However, the limitation within the ASV5 challenge's open track discarded the majority of the top performing readily available pretrained models. As a result, we first explored additional SSL model families trained only on the LibriSpeech [24] dataset, and benchmarked them as frontend feature extractors on a subset of the ASV5 data. We then selected the base variants of wavLM and wav2vec2 and finetuned their parameters for the deepfake detection task. Our final challenge submission aggregated the predictions of several pretrained and finetuned models and obtained a 17.08% EER.

## 2. Benchmarking pretrained model representations

We investigate how well pretrained audio representations transfer to the task of audio deepfake detection. We consider three classes of models. The first class are self-supervised models, which are pretrained on unlabelled data:

- DeCoAR2 [25] is a Deep Contextualized Acoustic Representation model using vector quantisation. The model was trained on 960h of LibriSpeech.
- HuBERT [11] is a Hidden-Unit BERT approach for self-supervised speech representation learning. It was trained on LibriSpeech and finetuned on different subsets.
- Distill-HuBERT [26] is a version of HuBERT pretrained SSL model, that reduces its size by 75%. This model was trained for several downstream tasks using SUPERB dataset.
- wavLM [12] learns masked speech prediction and it denoises the data during training to enhance the performance. The base version is trained on 960h of LibriSpeech data.
- wav2vec2.0 [10] is a well known framework for self-supervised learning of speech representations, it masks the speech input in the latent space and solves a contrastive task defined over a quantisation of the latent representations which are jointly learned. The base version was trained on LibriSpeech data.
- BEATs [27] is an iterative audio pretraining framework to learn Bidirectional Encoder representation from Audio Transformers, where an acoustic tokenizer and an audio SSL model are iteratively optimised. This model was trained on Audioset-2M [28] which also includes non-speech audio.

The second class of models are speaker embedding networks. We selected this category of models, since they were shown to capture other information as well [29]. We selected the ECAPA-TDNN [30] and TitaNet [31] models. ECAPA-TDNN is a time delay neural network that applies statistics pooling to project variable-length utterances into fixed-length speaker embeddings; this model was trained on the VoxCeleb dataset. For TitaNet we use its `large` variant, which was trained for speaker verification and diarisation on tens of thousands of hours of audio data from VoxCeleb 1, VoxCeleb 2, Fisher, Switchboard, LibriSpeech and SRE dataset.

Finally, the third class of models is that of learnable frontends. LEAF [32] was created to replace mel-filterbanks for audio classification of speech, music, audio events and animal sounds. HEAR’s YAMNet [33] is also trained for audio classification using a knowledge distillation approach with transformers and CNNs. Both models were trained on the Audioset data.

As topline, we also include results for two models trained on LibriLight [21] or Multilingual LibriSpeech [24], which were consequently not allowed in the challenge: `wavLM-large` and `wav2vec2-xls-r-2b`. The large variant of WavLM is a three times larger model than the base one, trained on 94k hours of speech (60k LibriLight, 10k Giga-Speech, 24k VoxPopuli). The XLS-R 2B variant of wav2vec [17] is a large-scale model for cross-lingual speech representation learning based on wav2vec 2.0. This model was trained on nearly half a million hours of publicly available speech audio in 128 languages.

To get a grasp of the models’ inherent capabilities, we plot t-SNE projections of the `wavLM-base` and `wav2vec2-xls-r-2b` representations for a subset of the ASV5 data (see Figure 1). It can be noticed that the

Table 1: Performance of self-supervised representations in terms of equal error rate (EER) on a subset of 27k samples from the ASV5 development set. Last two models are pretrained on either LibriLight or Multilingual LibriSpeech, and hence they do not adhere to the challenge rules. Lower values are better. The models are listed in decreasing order of their performance and have associated information regarding their parameter count and extracted feature’s dimension.

	<b>Model</b>	<i># param</i>	<i>feat dim</i>	EER ↓ [%]
1	LEAF [32]	4M	40	50.14
2	Distill-HuBERT [26]	23M	768	32.37
3	ECAPA-TDNN [30]	6M	192	28.23
4	HEAR’s YAMNet [33]	4M	184	23.69
5	TitaNet-large [31]	23M	192	20.84
6	BEATs [27]	90M	768	19.23
7	DeCoAR2 [25]	85M	768	18.74
8	HuBERT [11]	95M	768	16.47
9	wav2vec2-base [10]	94M	768	13.33
10	wavLM-base [12]	94M	768	<b>9.93</b>
Models pretrained on LibriLight or Multilingual LibriSpeech				
11	wavLM-large [12]	300M	1024	6.67
12	wav2vec2-xls-r-2b [17]	2B	1920	0.96

`wav2vec2-xls-r-2b` features exhibit a clear separation between the four subsets of data: spoofed samples from train (red), bonafide samples from train (green), spoofed samples from dev (purple), bonafide samples from dev (black). This indicates that this self-supervised representation is powerful enough to discriminate spoofed from bonafide samples even when simple backend classifiers are employed. Moreover, we observe distinct clusters even inside the spoofed data, presumably corresponding to each of the eight attacks in training and development sets. For the `wavLM-base` representation, some similar clusters emerge, but their separation hyperplanes are not as clearly defined. It is worth mentioning that the t-SNE projection is non-linear, and that the `wavLM` features may still exhibit linearly separable clusters in the  $N$ -dimensional space.

For audio deepfake detection benchmarking we selected two random subsets of 27k samples from each of the training and development sets of the ASV5 Challenge data, respectively. Both subsets have the same data statistics as the original sets, i.e., 8 spoof files to 1 bonafide file. We use the SSL models as feature extractors (without finetuning) and evaluate their performance for deepfake detection by linear probing. Specifically, we train a logistic regression model over the average pooled representations to predict the ‘bonafide’ or ‘spoof’ label. The logistic regression uses a regularisation term<sup>1</sup> of  $10^3$ .

Table 1 shows these results. We observe a wide range of performances, from 50% EER (random chance) to less than 10% EER, which is obtained by the base version of the `wavLM` model. Models pretrained on LibriLight (`wavLM-large`) or Multilingual LibriSpeech (`wav2vec2-xls-r-2b`) improve the performance even further to 0.96% EER for the later. But since these models were not allowed in the competition, we focus on the `wavLM-base` model and proceed to finetune its parameters on the audio deepfake task.

<sup>1</sup>As defined by the `C` parameter in the scikit-learn documentation

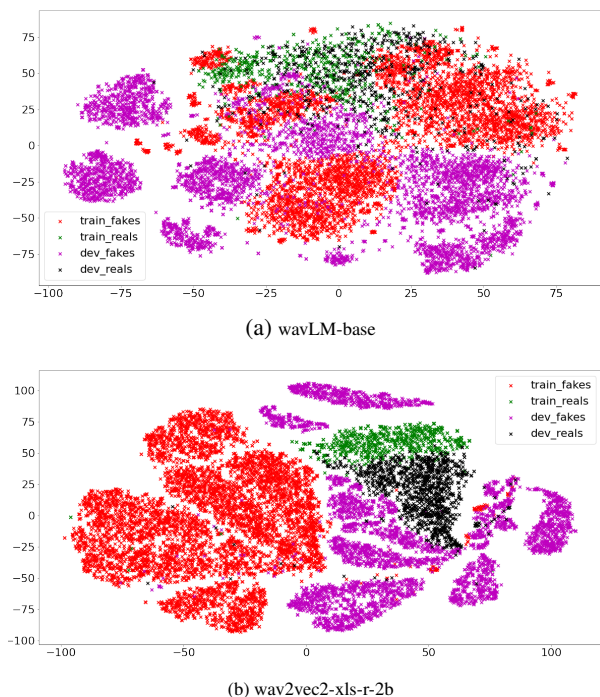


Figure 1: t-SNE plots of the pretrained (a) `wavLM-base`, (b) `wav2vec2-xls-r-2b` representations on a random subset of the ASV5 training and development data.

### 3. Model finetuning

In the previous section we found that the pretrained `wavLM-base` and `wav2vec2-base` models exhibited the best deepfake detection performance within the ASV5 challenge’s limitations. To further increase the models’ discriminative power, we perform a light finetuning of their weights. For this we also use external audio data and signal-based data augmentation. The finetuned models were again used as feature extractors, and a logistic regressor provided the final classification labels.

#### 3.1. Additional audio data and data augmentation

The initial phase of the ASV5 challenge released a set of 182,357 training samples (18,797 bonafide — 163,560 spoofed); 140,950 development samples (31,334 bonafide — 109,616 spoofed), and a small progress phase evaluation set of 40,765 samples (no labels were given for this subset). The final evaluation was performed over 680,774 samples. From our initial data analysis, we assumed that one of the challenge’s objectives was to explore the models’ generalisation abilities. Therefore, we also included in the training data a random selection of samples from other audio deepfake datasets, as follows:

- **ASVspoof 2019** (ASV19) [34] – from all subsets;
- **ASVspoof 2021** (ASV21) [35] – from the evaluation subset;
- **Fake or Real** (FoR) [36] – from the ‘norm’ subset;
- **In the Wild** (ITW) [37] – from all.

We also explored signal-based data augmentation. Specifi-

cally, we added white noise<sup>2</sup> and reverberation<sup>3</sup> to the bonafide files in ASV5; we applied a single random augmentation to an audio sample; half of the bonafide samples were augmented. We did not alter the spoofed samples.

Based on the above steps, we obtained three dataset variants which were used for finetuning:

- `medium-27k` consists of a subset of 27k samples from ASV5 training set having the same distribution as the original set (8 spoofed files to 1 bonafide file, equal number of files from each spoofing attack)—same as in Section 2;
- `augm-31k` consists of 31k samples: 13k samples from ASV5 with augmentation for half of the bonafides, 6.1k from ASV19, 8.6k from ASV21, 1.6k from ITW, 1.8k from FoR;
- `augm-114k` consists of 114k samples: 102k samples from ASV5 with augmentation for half of the bonafides, 2.9k from ASV19, 6.8k from ASV21, from 1.6k FoR, 1.6k from ITW.

The `augm-31k` was selected to have a similar number of samples as the `medium-27k` set. While the `augm-114k` subset was chosen to explore if more finetuning data increases the model’s ability to generate more discriminative features for the deepfake detection task.<sup>4</sup>

In the numeric evaluation we used the same 27k subset of the development set as in Section 2 and the small progress phase evaluation set provided by the organisers.

#### 3.2. Implementation details

The models were finetuned to minimise the binary cross-entropy loss over the deepfake detection task. The Adam optimiser was used with a learning rate of  $3 \cdot 10^{-5}$  and a linear scheduler with a warm up ratio of 0.1 over . We finetuned the models over 5 epochs using a batch size of either 8 (for `medium-27k` and `augm-114k`) or 16 (for `augm-31k`). Training was performed on a single Tesla V100 16GB GPU and took around 20 hours for the `augm-114k` split.

The finetuning process yielded 3 variants of the `wavLM-base` model corresponding to the three datasets described in the previous section, and one variant of the `wav2vec2-base` model finetuned only with the `medium-27k` subset.

#### 3.3. Results

The results are shown in Table 2 and indicate that finetuning improves over the pretrained representations. Among the three `wavLM-base` finetuning variants, the ones using augmented data perform best: the `augm-31k` set gives the lowest error on the development set (0.61% EER), while the `augm-114k` set gives the lowest error on the progress phase evaluation set (7.26% EER).

For the `wav2vec2-base` representation we observe similar improvements over the pretrained variant. However, the performance remains worse in the absolute than that of the `wavLM-base` model: 11.91% EER for the finetuned `wav2vec` model (row 6) versus 7.26% EER for the best finetuned `wavLM` model (row 4).

<sup>2</sup>With a signal to noise ratio of 25dB

<sup>3</sup>As described in the torchaudio tutorial: [https://pytorch.org/audio/stable/tutorials/audio\\_data\\_augmentation\\_tutorial.html](https://pytorch.org/audio/stable/tutorials/audio_data_augmentation_tutorial.html)

<sup>4</sup>Given our limited computational resources, we did not attempt to finetune the models using the complete datasets.

Table 2: EER [%] performance of finetuned wavLM and wav2vec2 models on the ASV5 development and progress phase evaluation sets.

Model type		Training set	EER[%] ↓	
			Dev	Prog
wavLM variants: wavlm-base				
1	Pretrained	–	9.93	15.82
2	Finetuned	medium-27k	4.16	–
3	Finetuned	augm-31k	<b>0.61</b>	9.02
4	Finetuned	augm-114k	2.97	<b>7.26</b>
wav2vec2 variants: wav2vec2-base				
5	Pretrained	–	13.33	–
6	Finetuned	medium-27k	5.85	11.91

Table 3: EER [%] performance on development, progress phase evaluation and final evaluation sets for the late fusion combinations of four wavLM-base models (see Table 2). The models vary by type (pretrained – PT or finetuned – FT) and data used for finetuning (medium-27k, augm-31k or augm-114k). Lower values are better. Best results are marked in boldface.

Type:	PT	FT	FT	FT	EER[%] ↓		
					Dev	Prog	Eval
Data:	–	27k	31k	114k			
1				✓	2.97	7.26	–
2	✓			✓	1.16	–	–
3		✓		✓	1.17	–	–
4			✓	✓	<b>0.56</b>	–	–
5		✓	✓	✓	0.60	–	–
6	✓	✓	✓	✓	0.72	<b>6.56</b>	<b>17.08</b>

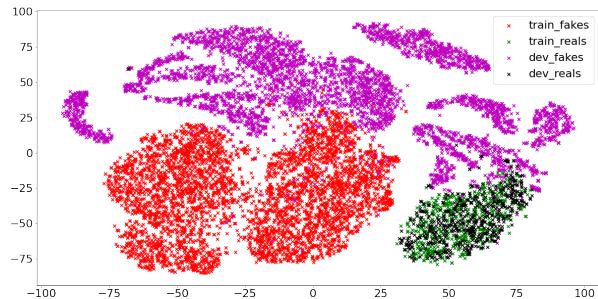
We again examined the t-SNE plots of the features extracted from the finetuned models and show them in Figure 2. It can be noticed that, as opposed to Figure 1a, the separation between the real and fake samples is more clearly defined. However, the different attacks still do not seem as clustered as for the wav2vec2-xls-r-2b features (see Figure 1b).

## 4. Model ensemble

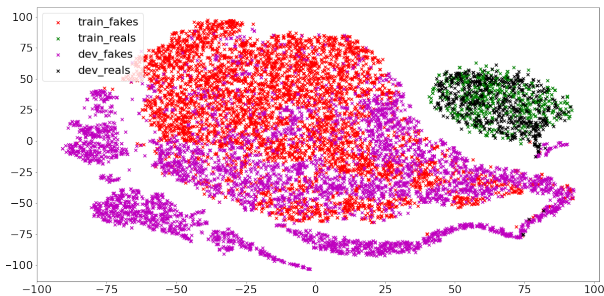
For the final submission to the ASV5 Challenge, we combined the predictions of multiple models using late fusion. We learn a set of weights using separate logistic regression over the probabilities output by the models presented in Section 3. The late fusion weights are unconstrained, so they can be either positive or negative. To learn the weights, we used the medium-27k dataset train split.

The results are shown in Table 3. We start from the best performing model found in the previous section (listed in row 1 of the table), and first combine it with one (rows 2–4), then two (row 5) and finally all of the others models (row 6). On the development set, we observe improvements by using model combinations, with the best combination of two consisting of the models finetuned on the largest datasets: augm-31k and augm-114k (row 4). Our submission to the challenge corresponds to the combination of all four variants (row 6) and yields an EER of 6.56% on the progress phase evaluation set, and 17.08% on the final evaluation set.<sup>5</sup>

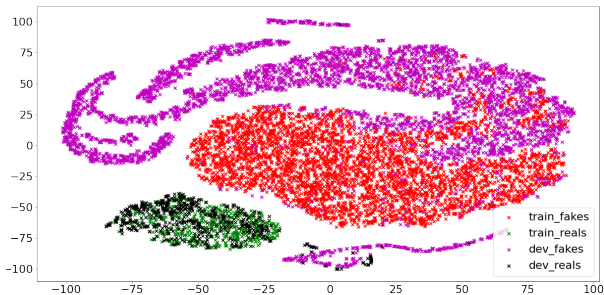
<sup>5</sup>Not all results over the progress phase evaluation set could be presented here, as the phase was closed by the time of the current submission.



(a) wavLM-finnetuned on medium-27k



(b) wavLM-finnetuned on augm-31k



(c) wavLM-finnetuned on augm-114k

Figure 2: t-SNE plots of the representations obtained from the finetuned wavLM models using the (a) medium-27k, (b) augm-31k, (c) augm-114k datasets.

## 5. Conclusions

This paper presented our submission to the ASVspooF 2024 deepfake detection challenge. First, we benchmarked a set of ten pretrained representations belonging to three classes of models (self-supervised, speaker embedding, learnable frontends) and have shown that the self-supervised models outperform the others, with wavLM achieving the best performance. Second, we have shown that we can further improve the performance of the pretrained representations by finetuning them for the task of deepfake detection. When finetuning we found data augmentation to be an important component. Our final submission combined four different models (pretrained and finetuned) in an ensemble and delivered the best performance.

## 6. Acknowledgements

This work was funded by EU Horizon projects AI4TRUST (No. 101070190) and by CNCS/CCCDI UEFISCDI (No. PN-IV-P8-8.1-PRE-HE-ORG-2023-0078).

## 7. References

- [1] X. Wang and J. Yamagishi, “Investigating self-supervised front ends for speech spoofing countermeasures,” in *Proc. Odyssey: The Speaker and Language Recognition Workshop*, 2022.
- [2] J. M. Martín-Doñas and A. Álvarez, “The Vicomtech audio deepfake detection system based on wav2vec2 for the 2022 ADD challenge,” in *Proc. ICASSP*, 2022, pp. 9241–9245.
- [3] H. Tak, M. Todisco, X. Wang, J. Jung, J. Yamagishi, and N. W. D. Evans, “Automatic speaker verification spoofing and deepfake detection using wav2vec 2.0 and data augmentation,” in *Proc. Odyssey: The Speaker and Language Recognition Workshop*, 2022.
- [4] Y. Yang, H. Qin, H. Zhou, C. Wang, T. Guo, K. Han, and Y. Wang, “A robust audio deepfake detection system via multi-view feature,” in *Proc. ICASSP*, 2024, pp. 13 131–13 135.
- [5] Y. Guo, H. Huang, X. Chen, H. Zhao, and Y. Wang, “Audio deepfake detection with self-supervised wavLM and multi-fusion attentive classifier,” in *Proc. ICASSP*, 2024, pp. 12 702–12 706.
- [6] H. Wu, J. Zhang, Z. Zhang, W. Zhao, B. Gu, and W. Guo, “Robust spoof speech detection based on multi-scale feature aggregation and dynamic convolution,” in *Proc. ICASSP*, 2024, pp. 10 156–10 160.
- [7] S. Saha, M. Sahidullah, and S. Das, “Exploring green AI for audio deepfake detection,” *CoRR*, vol. abs/2403.14290, 2024.
- [8] O. Pascu, A. Stan, D. Oneata, E. Oneata, and H. Cucu, “Towards generalisable and calibrated synthetic speech detection with self-supervised representations,” in *Proc. Interspeech*, 2024.
- [9] R. Balestrieri, M. Ibrahim, V. Sobal, A. Morcos, S. Shekhar, T. Goldstein, F. Bordes, A. Bardes, G. Mialon, Y. Tian, A. Schwarzschild, A. G. Wilson, J. Geiping, Q. Garrido, P. Fernandez, A. Bar, H. Pirsivash, Y. LeCun, and M. Goldblum, “A cookbook of self-supervised learning,” *CoRR*, vol. abs/2304.12210, 2023. [Online]. Available: <https://doi.org/10.48550/arXiv.2304.12210>
- [10] A. Baevski, Y. Zhou, A. Mohamed, and M. Auli, “Wav2vec 2.0: A framework for self-supervised learning of speech representations,” in *Proc. NeurIPS*, 2020.
- [11] W. Hsu, B. Bolte, Y. H. Tsai, K. Lakhota, R. Salakhutdinov, and A. Mohamed, “HuBERT: Self-supervised speech representation learning by masked prediction of hidden units,” *IEEE ACM Trans. Audio Speech Lang. Process.*, vol. 29, pp. 3451–3460, 2021.
- [12] S. Chen, C. Wang, Z. Chen, Y. Wu, S. Liu, Z. Chen, J. Li, N. Kanda, T. Yoshioka, X. Xiao, J. Wu, L. Zhou, S. Ren, Y. Qian, Y. Qian, J. Wu, M. Zeng, X. Yu, and F. Wei, “WavLM: Large-scale self-supervised pre-training for full stack speech processing,” *IEEE J. Sel. Top. Signal Process.*, vol. 16, no. 6, 2022.
- [13] S. Yang, P. Chi, Y. Chuang, C. J. Lai, K. Lakhota, Y. Y. Lin, A. T. Liu, J. Shi, X. Chang, G. Lin, T. Huang, W. Tseng, K. Lee, D. Liu, Z. Huang, S. Dong, S. Li, S. Watanabe, A. Mohamed, and H. Lee, “SUPERB: Speech processing universal performance benchmark,” in *Proc. Interspeech*, 2021, pp. 1194–1198.
- [14] X. Chen, Y. Zhang, G. Zhu, and Z. Duan, “UR channel-robust synthetic speech detection system for ASVspoof 2021,” *CoRR*, vol. abs/2107.12018, 2021. [Online]. Available: <https://arxiv.org/abs/2107.12018>
- [15] H. Tak, J. Patino, M. Todisco, A. Nautsch, N. Evans, and A. Larcher, “End-to-end anti-spoofing with RawNet2,” in *Proc. ICASSP*, 2021.
- [16] A. Conneau, A. Baevski, R. Collobert, A. Mohamed, and M. Auli, “Unsupervised cross-lingual representation learning for speech recognition,” in *Proc. Interspeech*, 2021.
- [17] A. Babu, C. Wang, A. Tjandra, K. Lakhota, Q. Xu, N. Goyal, K. Singh, P. von Platen, Y. Saraf, J. Pino, A. Baevski, A. Conneau, and M. Auli, “XLS-R: Self-supervised cross-lingual speech representation learning at scale,” in *Proc. Interspeech*, 2022.
- [18] C. Wang, J. Yi, J. Tao, H. Sun, X. Chen, Z. Tian, H. Ma, C. Fan, and R. Fu, “Fully automated end-to-end fake audio detection,” in *Proc. International Workshop on Deepfake Detection for Audio Multimedia*, 2022, pp. 27–33.
- [19] Y. Xie, H. Cheng, Y. Wang, and L. Ye, “Learning a self-supervised domain-invariant feature representation for generalized audio deepfake detection,” in *Proc. Interspeech*, 2023, pp. 2808–2812.
- [20] X. Wang, H. Delgado, H. Tak, J.-w. Jung, H.-j. Shim, M. Todisco, I. Kukanov, X. Liu, M. Sahidullah, T. Kinunen, N. Evans, K. A. Lee, and J. Yamagishi, “ASVspoof 5: Crowdsourced speech data, deepfakes, and adversarial attacks at scale,” in *ASVspoof Workshop 2024 (accepted)*, 2024.
- [21] J. Kahn, M. Rivière, W. Zheng, E. Kharitonov, Q. Xu, P. Mazaré, J. Karadayi, V. Liptchinsky, R. Collobert, C. Fuegen, T. Likhomanenko, G. Synnaeve, A. Joulin, A. Mohamed, and E. Dupoux, “Libri-light: A benchmark for ASR with limited or no supervision,” in *Proc. ICASSP*, 2020, pp. 7669–7673.
- [22] V. Pratap, Q. Xu, A. Sriram, G. Synnaeve, and R. Collobert, “MLS: A large-scale multilingual dataset for speech research,” in *Proc. Interspeech*, 2020, pp. 2757–2761.
- [23] D. Snyder, G. Chen, and D. Povey, “MUSAN: A Music, Speech, and Noise Corpus,” 2015, arXiv:1510.08484v1.
- [24] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, “Librispeech: An ASR corpus based on public domain audio books,” in *Proc. ICASSP*, 2015, pp. 5206–5210.
- [25] S. Ling and Y. Liu, “DeCoAR 2.0: Deep contextualized acoustic representations with vector quantization,” *CoRR*, vol. abs/2012.06659, 2020.
- [26] H.-J. Chang, S.-w. Yang, and H.-y. Lee, “DistilHuBERT: Speech representation learning by layer-wise distillation of hidden-unit BERT,” in *Proc. ICASSP*, 2022, pp. 7087–7091.
- [27] S. Chen, Y. Wu, C. Wang, S. Liu, D. Tompkins, Z. Chen, W. Che, X. Yu, and F. Wei, “BEATs: Audio pre-training with acoustic tokenizers,” in *Proc. ICML*, vol. 202, 2023, pp. 5178–5193.
- [28] J. F. Gemmeke, D. P. W. Ellis, D. Freedman, A. Jansen, W. Lawrence, R. C. Moore, M. Plakal, and M. Ritter, “Audio set: An ontology and human-labeled dataset for audio events,” in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2017, pp. 776–780.

- [29] A. Stan, “Residual information in deep speaker embedding architectures,” *Mathematics*, vol. 10, no. 21, 2022. [Online]. Available: <https://www.mdpi.com/2227-7390/10/21/3927>
- [30] B. Desplanques, J. Thienpondt, and K. Demuynck, “ECAPA-TDNN: Emphasized channel attention, propagation and aggregation in TDNN based speaker verification,” in *Proc. Interspeech*, 2020, pp. 3830–3834.
- [31] N. R. Koluguri, T. Park, and B. Ginsburg, “TitaNet: Neural model for speaker representation with 1d depth-wise separable convolutions and global context,” in *Proc. ICASSP*, 2022, pp. 8102–8106.
- [32] N. Zeghidour, O. Teboul, F. de Chaumont Quitry, and M. Tagliasacchi, “LEAF: A learnable frontend for audio classification,” in *Proc. ICLR*, 2021.
- [33] J. Turian, J. Shier, H. R. Khan, B. Raj, B. W. Schuller, C. J. Steinmetz, C. Malloy, G. Tzanetakis, G. Velarde, K. McNally, M. Henry, N. Pinto, C. Noufi, C. Clough, D. Herremans, E. Fonseca, J. H. Engel, J. Salamon, P. Esling, P. Manocha, S. Watanabe, Z. Jin, and Y. Bisk, “HEAR: Holistic evaluation of audio representations,” in *Proc. NeurIPS Competitions and Demonstrations*, vol. 176, 2021, pp. 125–145.
- [34] X. Wang *et al.*, “ASVspooF 2019: A large-scale public database of synthesized, converted and replayed speech,” *Comput. Speech Lang.*, vol. 64, 2020.
- [35] X. Liu, X. Wang, M. Sahidullah, J. Patino, H. Delgado, T. Kinnunen, M. Todisco, J. Yamagishi, N. W. D. Evans, A. Nautsch, and K. A. Lee, “ASVspooF 2021: Towards spoofed and deepfake speech detection in the wild,” *IEEE ACM Trans. Audio Speech Lang. Process.*, vol. 31, pp. 2507–2522, 2023.
- [36] R. Reimao and V. Tzerpos, “FoR: A dataset for synthetic speech detection,” in *Proc. SpeD*, 2019.
- [37] N. M. Müller, P. Czempin, F. Dieckmann, A. Froghyar, and K. Böttinger, “Does audio deepfake detection generalize?” in *Proc. Interspeech*, 2022.